COURSE DESCRIPTION CARD - SYLLABUS

Course name
Microcontroller systems and programming [S1Mech2>SMiP1]

Course

Field of study
Mechatronics

Year/Semester
2/4

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
compulsory

Number of hours

Lecture
15

Laboratory classes
15

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

2,00

Coordinators

mgr inż. Daniel Wyrwał
daniel.wyrwal@put.poznan.pl

Lecturers

Prerequisites

Basic knowledge of automation, computer science, electronics, digital systems, and fundamentals of C programming.

Course objective

Familiarizing students with the structure, operation, and programming of simple microcontrollers, as well as designing and programming microprocessor controllers, including control systems for mechatronic devices using Arduino libraries.

Course-related learning outcomes

Knowledge:
Understanding the basic structure and operation of microcontrollers and microprocessor controllers, their capabilities, and technical parameters.
Handling ports, ADC converters, timers, timing circuits, and serial transmission using Arduino libraries.
Knowledge of interrupt handling and integration of microcontrollers with controllers using Arduino libraries.

Skills:
Designing control systems based on 8-bit and 32-bit microcontrollers.
Programming microcontrollers in C, including input/output handling, interrupts, and serial communication using Arduino libraries.
Implementing communication methods between devices using Arduino libraries.

Social competences:
Understanding the need for continuous learning and keeping up with technological advancements.
Awareness of the significance of microprocessor systems for the economy and society.
Ability to organize teamwork in embedded system projects.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Written test: Open-ended and multiple-choice questions to assess theoretical knowledge.
Laboratory assessment: Verifying students' ability to write and upload programs to a microcontroller.
Project assessment: Evaluation of documentation and implementation of microprocessor controller projects.

Programme content

Structure and programming of microcontrollers and microprocessor controllers.
Ports, peripheral connection, configuration, and programming.
Variable declaration, conditional statements, and loops in control systems.
Serial communication (UART, SPI, I2C) - basics of programming.
Timers, timing circuits, and PWM control.
ADC converters: application and programming.
Interrupt handling in microcontrollers and microprocessor controllers.
Designing microprocessor controllers for mechatronic systems.

Course topics

Theoretical lectures and presentations.
Laboratory exercises covering microcontroller programming.
Projects focused on implementing control systems.

Teaching methods

Lectures with multimedia presentations.
Practical laboratory sessions in microcontroller programming.
Individual and group projects.

Bibliography

Basic:
1. Official Arduino Documentation - Arduino.cc (docs.arduino.cc)
2. Covers the syntax of C/C++ used in Arduino, function descriptions, library explanations, and example projects.

Additional:
1. "AVR Microcontrollers in C: Basics of Programming" - Mirosław Kardaś
2. Introduction to programming AVR microcontrollers in C, covering microcontroller configuration, peripheral handling, and use of compilers and development environments.
3. "Symphony C++" - Jerzy Grębosz
4. A comprehensive introduction to C++, covering fundamental and advanced programming concepts. Useful for transitioning from C to object-oriented programming.

Breakdown of average student's workload

	Hours	ECTS
Total workload	50	2,00
Classes requiring direct contact with the teacher	30	1,00
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	20	1,00